

Тема: D.22.

1817 MU-1

Задача №17.

a) 1) Train::Train(const Train& train){  
    id = train.id  
    for (list<string>::iterator st =  
        train.stations.begin(); train  
            st != train.stations.end();  
            st++) {  
        stations.push\_back(\*st);  
    }  
}

2) bool Train::operator[](const string& name){  
    ~~if (find(stations.begin(), stations.end(), name~~  
    ; if (find(stations.begin(), stations.end(), name  
        != stations.end())) {  
        return true; } // не совсем кратко, но  
                            именно требуется воз-  
                            можность, по-  
        return false; } // тем, что возвращает bool-естествен-  
                            но.  
} // find() from algorithm

~~3) void Train::DeleteStation(const string& name){~~  
    ~~if {~~  
    for (auto st = stations.begin(); st != stations.end();  
        st++) {  
        if (\*st == name) {  
            stations.erase(st);  
            break;  
        }  
    }  
}

```

b) 1) void PrintMonth30(constmap<string, int> & year) {
    for (map<string, int>::iterator it = year.begin();
        it != year.end();
        it++) {
        if ( (*it).second == 30 ) {
            cout << (*it).first << endl;
        }
    }
}

```

```

2) void PrintMonth31(const map<string, int> & year) {
for (auto & elem : year)
for (const
    for (auto & elem : year) {
        if ( [elem] ( ) { return elem.second == 31; }
        {
            cout << elem.first << endl;
        }
    }
}

```



```

c) #include <thread>
    #include <mutex>
    #include <iostream>
    #define MAX 100000

```

```

using namespace std;

```

```

mutex mtx;

```

```

void PrintEven() { // (*)
    for (int i = 0; i < MAX; i += 2) {
        lock_guard<mutex> lock(mtx);
        cout << i << " is even" << endl;
    }
}

```

```

void PrintOdd() { // (*)
    for (int i = 1; i < MAX; i += 2) {
        lock_guard<mutex> lock(mtx);
        cout << i << " is odd" << endl;
    }
}

```

```

int main() {
    thread t1(PrintEven);
    thread t2(PrintOdd);

```

```

    t1.join(); // Если же в данной записи требуется
    t2.join(); // попеременно вывод, сначала один,
    return 0; // а потом другой, то все

```

```

} // конструируем lock-guard... нужно
    // разместить в местах, отмеченных (*).

```